Evaluation and Comparison of Anomaly Detection Algorithms in Annotated Datasets from the Maritime Domain

Mathias Anneken*, Yvonne Fischer*, Jürgen Beyerer*†
*Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB)
Karlsruhe, Germany
†Vision and Fusion Laboratory, Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
Email: {mathias.anneken, yvonne.fischer, juergen.beyerer}@iosb.fraunhofer.de

***Abstract*—Anomaly detection supports human decision makers in their surveillance tasks to ensure security. To gain the trust of the operator, it is important to develop a robust system, which gives the operator enough insight to take a rational choice about future steps. In this work, the maritime domain is investigated. Here, anomalies occur in trajectory data. Hence, a normal model for the trajectories has to be estimated. Despite the goal of anomaly detection in real life operations, until today, mostly simulated anomalies have been evaluated to measure the performance of different algorithms. Therefore, an annotation tool is developed to provide a ground truth on a non-simulative dataset. The annotated data is used to compare different algorithms with each other. For the given dataset, first experiments are conducted with the Gaussian Mixture Model (GMM) and the Kernel Density Estimator (KDE). For the evaluation of the algorithms, precision, recall, and f1-score are compared.**

***Keywords*—Anomaly detection; Gaussian Mixture Model; Kernel Density Estimation; Maritime domain**

I. Introduction

For surveillance tasks, the detection of abnormal behaviour is of utter importance. It helps operators to gain better situation awareness by focusing their attention on important events, while providing enough information to support the decision on the next possible steps. As stated by Fischer and Beyerer [1], the biggest challenge for advanced surveillance systems is not to collect as much data as possible, but rather to support a human decision maker by providing only the important information. Due to the large amount of recorded data, an operator can easily loose focus on the relevant events or even overlook them. In this regard, the detection of anomalies can help to reduce the amount of information, and hence help to distinguish crucial events from normal behaviour. A system will only provide such benefit, if the operator can trust in it. It is crucial for system acceptance by the human operator, that the surveillance system takes decisions on anomalies in a transparent way. Otherwise, an operator might not even consider the support of the anomaly detection, or even completely abandon the whole supporting system.

In order to achieve a transparent and reliable proposition, a surveillance system has to be designed with real life data in mind. By only using simulated data, a system might not be able to find anomalies in real data in an appropriate way, because it was only tested with abnormal behaviour as seen by the system designer and not with actual anomalies.

In this paper, the focus is mainly on anomaly detection in the maritime domain. Anomalies can be seen, e.g., in a deviation from normal sea lanes, a compared to the normal traffic wrong driving direction on a sea lane or too fast movement. Hence, algorithms for anomaly detection are investigated, which are able to detect anomalies in spatio-temporal data in form of trajectories.

The paper is structured as follows. Section II gives an overview about related work. Section III describes the used dataset and the annotating process. Section IV describes the designed test set-up with brief explanations of the used algorithms. In Section V, an empirical evaluation of the algorithms is conducted by using two subsets of prior described dataset. Section VI and VII give a conclusion and provide ideas about possible future work.

II. Related Work

Anomaly detection is important in different fields of application and research. A wide overview about different tasks and algorithms is given, e.g., by Chandola et al. [2] or for vision-based algorithms by Morris and Trivedi [3]. The applications vary from cyber security over fraud detection to image processing and sensor networks. Depending on the specific use case, an appropriate algorithm is chosen. These algorithms may base on different concepts, e.g., classification-based, clustering-based or nearest neighbour-based. Each of these algorithms has advantages and disadvantages. Thus, there is no all-in-one solution to cope with every problem.

Another type of classification is based on the ability of an algorithm to learn a model supervised, unsupervised or reinforced. For the first, the model is learned by using annotated data, i.e., an expert decides about anomalies in the training dataset. For unsupervised learning, there is no annotated data available. Therefore, an algorithm has to find patterns on its own. For the last type, an algorithms maximises the expected return for a specific application by following a prior specified policy.

In the field of maritime surveillance, several different approaches were introduced to identify abnormal behaviour of vessels, see, e.g., [4], [5], [6], [7], [8], [9].

Laxhammar et al. [4] compare the Kernel Density Estimation (KDE) and the Gaussian Mixture Model (GMM). They train the models with real life data and evaluate the performance of the algorithms with artificial anomalies. In order to evaluate the models' performances to reproduce the normal behaviour, the log-likelihood is calculated and compared using the median log-likelihood and the 1st percentile. For the anomaly detection performance, the needed number of observations for detecting an anomaly is compared.

Brax and Niklasson [5] introduce a state-based anomaly detection with discrete states for *heading*, *speed*, *position* and *relative position to the next vessel*. With these states, the probabilities for different roles (here called agents, e.g., raid agent, smuggler agent or friendly agent) are calculated. For the evaluation, different scenarios resembling specific situations are generated to obtain an accurate ground truth. The algorithms are only tested using this simulated ground truth.

Andersson and Johansson [6] use a Hidden Markov Model (HMM) to detect abnormal behaviour. The aim is to detect pirate attacks. They train the HMM with simulated data describing the normal behaviour of ships in a certain area. The data is divided into discrete states resembling the change of specific values, i.e., *distance to other objects*, *vessel size*, *identification number*, *speed* and *heading*. Afterwards, the model is evaluated by using a simulated pirate attack.

Laxhammer and Falkman [7] introduce the sequential conformal anomaly detection. This algorithm is based on the conformal prediction framework, which is explained, e.g., by Schafer and Vovk [10]. The main idea is to provide a reliable lower boundary for the probability that the prediction is correct. Therefore, the probability that the prediction is not correct is given as one parameter. For calculating the similarity between two trajectories, the Hausdorff distance is used. For the evaluation, a non-simulated dataset is used for training and anomalies are simulated.

Guillarme and Lerouvreur [8] introduce a novel approach. They first partition the training trajectories into stops and moves using the Clustering-Based Stops and Moves of Trajectories algorithm. Afterwards, the sub-trajectories are clustered using a similarity measure and the OPTICS algorithm, which utilizes the data density for clustering. The used clusters are picked for their quality by hand. With this results, motion patterns and junctions for the trajectories are defined. For testing purposes, they use historical satellite AIS data. The model is trained on this dataset and some results of the anomaly detection are shown. They give no information about the performance of the algorithm compared to other algorithms.

Fischer et al. [9] present an approach based on dynamic Bayesian networks. They model the relationship between situations in a situational dependency network. With this method, the probability of a prior defined situation, e.g., *a suspicious incoming smuggling vessel*, can be estimated. Thus, unusual behaviour of a vessel can be detected. This algorithm is tested with simulated data.

The main difference between the previous work in the maritime domain and this paper is the used dataset. Previous attempts relay either completely on simulated data ([6] and [9]) or at least the anomalies are created artificially ([4] and [7]), while here an annotated dataset based on real data is used. For the annotation, a tool to identify anomalies in a dataset and to annotated the anomalies is developed. The tool and the used dataset are described in the following section.

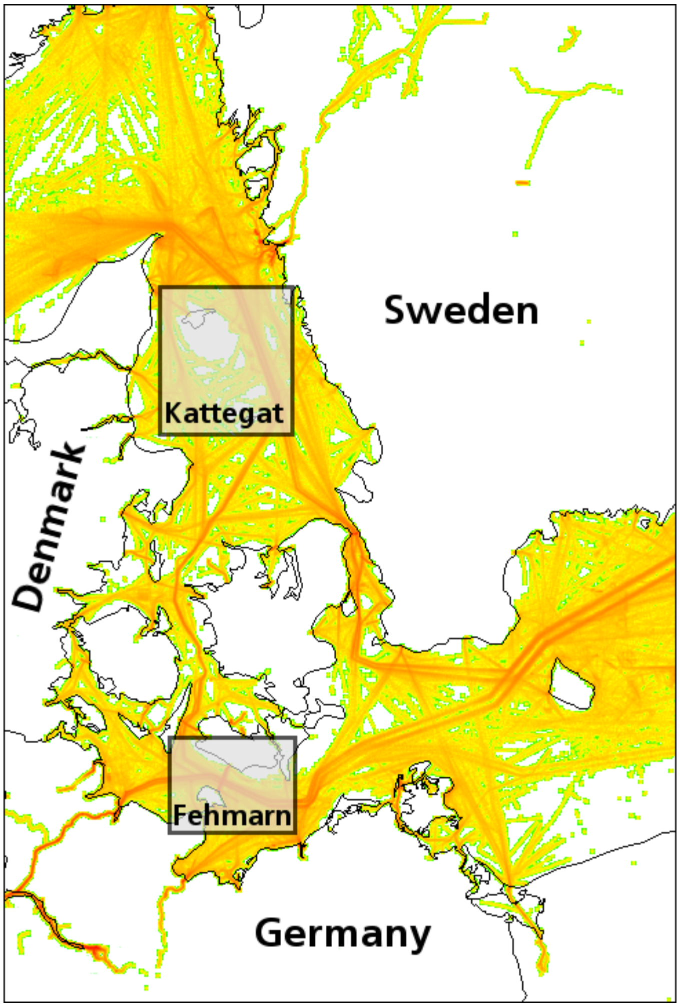


Fig. 1. Heat-map of the vessel traffic in the dataset. The traffic density is encoded by the colour, whereas the gradient from red to green represents the gradient from high density to low density. The marked areas (namely Fehmarn and Kattegat) are further analysed.

III. Dataset Annotation

As mentioned before, a dataset from the maritime domain is used. It was recorded using the automatic identification system (AIS). The AIS provides different kinds of data like *navigation-status*, *estimated time of arrival*, ..., *destination*. For the analysis only a subset of the whole available data is used, namely *position*, *speed*, *heading*, *maritime mobile service identity (MMSI)*, *timestamp* and *vessel-type*. Altogether, more than 2.4 million unique measurements are recorded.

Fig. 1 shows a heat-map representing all vessel positions in the dataset. The map encodes the traffic density with colours ranging from green for low density to red for high density. Regions with high traffic density can be identified as sea lanes

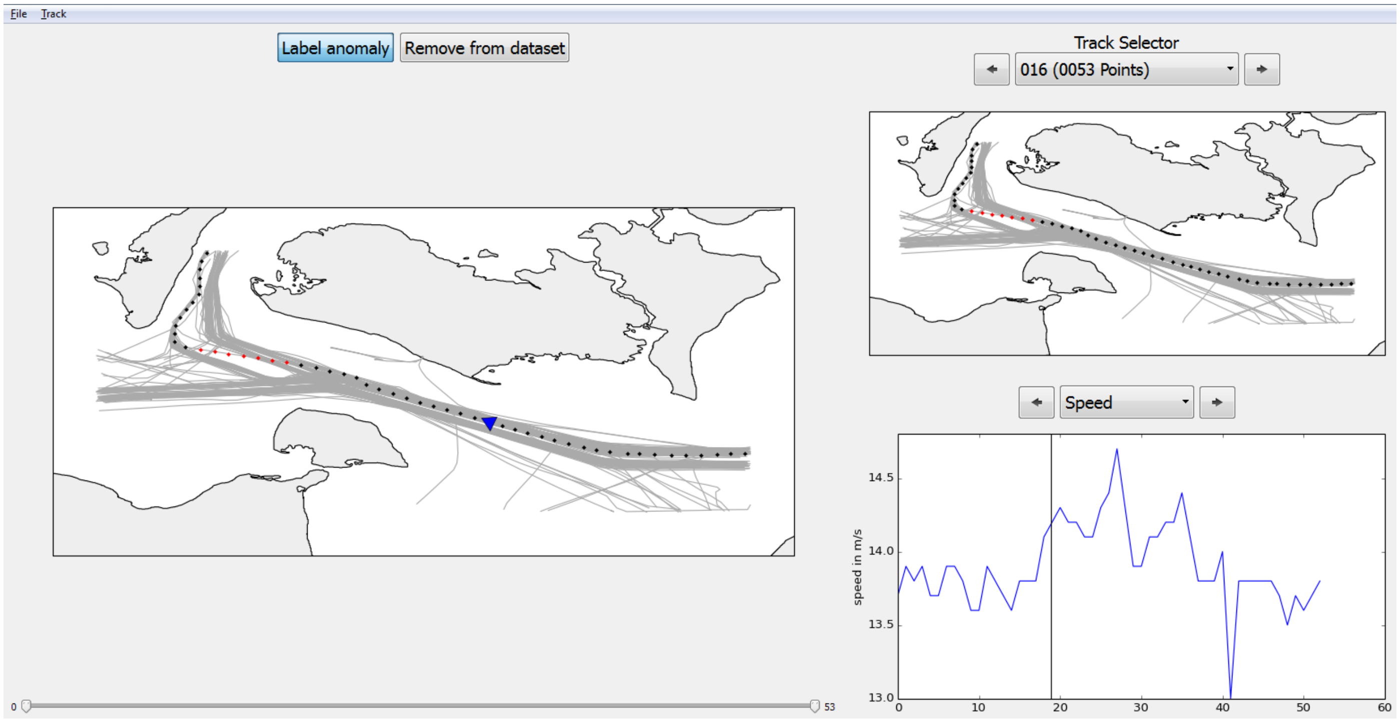


Fig. 2. The annotation tool: The map on the left side is used for annotating data. The right side is used to give an overview and a better understanding of the current situation, e.g., the heading. The grey polygons in the maps are landmasses and the white background is water. The grey lines represent all trajectories in the dataset (here tankers). The trajectory to annotate is marked with either black, red or blue dots. Black dots represent normal behaviour and red dots abnormal. The blue dots are removed from the dataset (not seen in this figure). The blue triangle depicts the position marked in the feature plot in the bottom right.

or harbours. Geographically, the recorded area comprises the western parts of the Baltic Sea, the Kattegat and parts of the Skagerrak. Temporally, it spans a whole week starting from 16th May 2011. Altogether, 3702 different vessels (unique MMSIs) grouped into 30 different vessel types were detected. In the first step, clearly wrong measurements as well as measurements generated by offshore structures (e.g., lights) are removed. Afterwards, 3550 unique vessels remain. For further processing the data points created by each ship are grouped by their corresponding MMSI and connected to tracks. If the time between two measurements with the same MMSI is too large (here, larger than 30 minutes), the track of the ship is split. Therefore, each vessel can generate more than one track and altogether 25 918 different tracks are detected.

For further investigations, only cargo vessels and tankers are used. In the prepared dataset, there are 1087 cargo vessels and 386 tankers. The two types have similar movement behaviour; therefore, they are treated as one. This means, there will always be one compound model for both types instead of a single one for each.

In order to detect non-simulated anomalies, a ground truth for the dataset is needed. Therefore, the dataset needs to be annotated. To achieve this, an annotation tool is implemented in Python with QT for the graphical user interface (GUI). The GUI is shown in Fig. 2. The left part of the window shows a map of the regarded area. The light grey polygons represent landmasses, the white background is equivalent to the sea, in this example an area around the islands Fehmarn and Lolland. The grey lines indicate all relevant tracks in the chosen area (here, only the tankers are shown), and the black and red dots represent the track which is currently being annotated. In the top right of the window, a track that needs to be annotated can be selected by using the arrows or the drop-down menu. Below this, an overview map is shown with the same colours as in the main map. This map always shows the whole area, while the main map can be used to get a zoomed view of a specific area. In the bottom right, a plot of a specific time-dependent feature can be seen. Here, the speed in meter per second is displayed for the selected track.

The black vertical line in the feature plot is connected with the blue triangle in the main map. The blue triangle shows both position and heading of the vessel at the time specified by the black line. This provides a better understanding of the vessel situation. Thus it helps the annotator to decide, whether abnormal data is on hand or not. In order to annotate points as anomalies, a rectangle selector can be drawn in the main map, and all points inside the selector will be annotated as abnormal.

Furthermore, a rectangle selector can be used either on the main map or on the overview map to get a zoomed view of the specified area in the main map. In order to show only a part of the whole track, the slider under the main map can be used; alternatively, a part of the track can be selected in the feature plot.

Due to the huge amount of data and the effort that would be necessary to annotate the whole dataset, the amount of data for the first experiments is narrowed down. Hence, two areas are chosen which reflect specific criteria. The trajectories of the normal behaviour for the chosen areas is depicted in Fig. 3 and Fig. 4. In figure 3 the Kattegat area is depicted (which

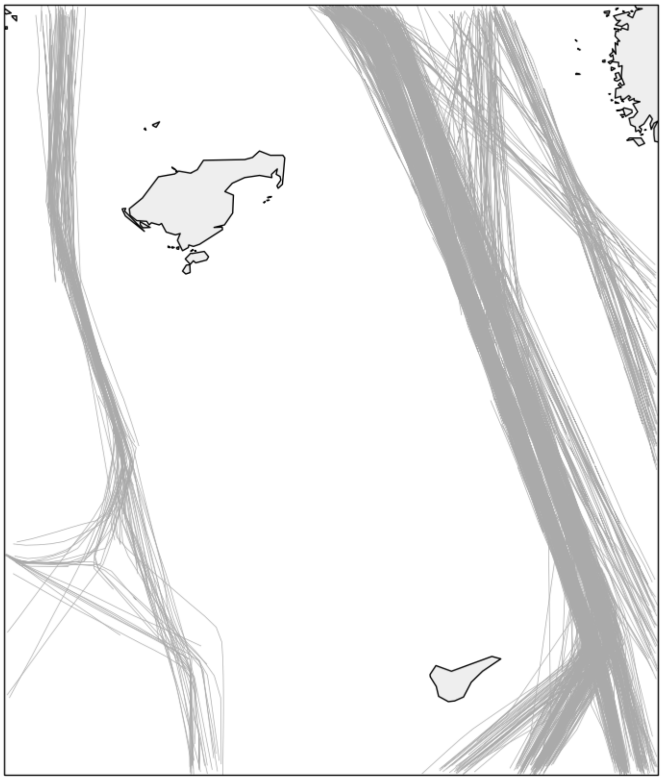


Fig. 3. Normal trajectories (grey lines) in the annotated and evaluated areas around the Kattegat. The light grey polygons represent landmasses, the white background represents represents water.

is actually only a part of the Kattegat). In this area, there are no ports called at by any of the tankers or cargo vessels in the dataset. Because of the many shoals around the two bigger islands (in the north Læsø and in the south Anholt), there are only a few main lanes where most of the vessels travel. These lanes are going from north to south or vice versa. Vessels travelling from west to east are considered as anomaly. In total, there are 516 different tankers and cargo vessels in this area creating 30 531 data points.

Fig. 4 shows the Fehmarn area where the situation is more complex. There is one sea lane between the islands Fehmarn and Lolland. Furthermore, there is a sea lane for each direction from west to east. In the western part, these lanes split into two lanes for each direction, one going north, and one going west. All vessels coming from the south are marked as anomalies, even though some of them might be normal traffic going to ports in Rostock or other German cities. Due to the small amount of vessels taking this way, normal and abnormal traffic cannot be distinguished proficiently. Therefore, the tracks are annotated as anomalies. Furthermore, there are several ports which are called at by the selected vessels in this area. These are marked with red dots in the figure. Thus, there is a higher percentage of abnormal data in this region. Altogether, there are 602 unique tankers and cargo vessels in the two regions generating 36 069 data points.

With the above described tool, the tanker and cargo vessels in the two areas can be annotated. For the Fehmarn area, 14.5% of the data points by cargo vessels and 8.7% of the data points by tankers are marked as unusual ones. Vessels that are moored in a harbour are removed from the dataset, for the behaviour in harbours is out of scope for this work. In the Kattegat area, 5.4% of the data points by tankers and 6.6% of the data points by cargo vessels were annotated as abnormal.

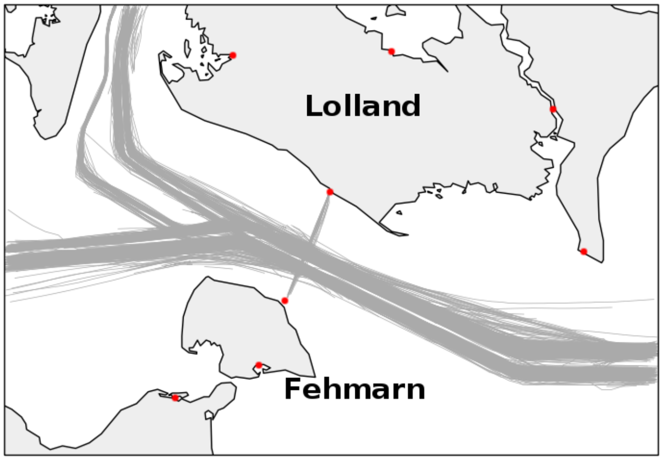


Fig. 4. Normal trajectories (grey lines) in the annotated and evaluated areas around Fehmarn. The light grey polygons represent landmasses, the white background represents water, the red dots represent harbours.

IV. Test Set-Up

Here, the two algorithms GMM and KDE are evaluated. Therefore, the algorithms, the possible parameters to set, the metrics, and the detection of anomalies with these algorithms are introduced. Afterwards, the optimal parameters for the given areas of the dataset are estimated. The observed features for the models are the position in latitude and longitude as well as the speed vector split into its latitude and longitude parts. This results in the feature vector $x_i = \{x_{\text{lat}}, x_{\text{lon}}, v_{\text{lat}}, v_{\text{lon}}\}$ for each data point i. For each area, a different model has to be trained. In order to generate a better model of a specific area, it can be divided by a grid, and each grid-cell is assigned its own model. The grid-size has to be estimated as well as the parameters of the models. For these purposes, the Python package *scikit-learn* [11] is used.

A. Algorithms

1) Gaussian Mixture Model: A GMM consists of n components. Each component is a multivariate normal distribution with the mean vector μ and the covariance matrix Σ. Together, they form the parameter set $\theta_i = \{\mu_i, \Sigma_i\}$ for each component i. The dimension of μ and Σ depend on the number of observed features k. The probability density function is given by

$$f(x) = \sum_{i=1}^{n} f_g(x, \theta_i)$$

with the density function for each component given by

$$f_g(x, \theta_i) = \frac{1}{(2\pi)^{\frac{k}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_i)^T {\Sigma_i}^{-1} (x-\mu_i)\right).$$

In order to estimate the parameter sets θ_i, the expectation-maximisation (EM) algorithm is used. Prior to this estimation, the number of components n must be available. More details on the GMM and the EM algorithm is given, e.g., by Barber [12].

2) Kernel Density Estimator: The KDE or Parzen-window density estimation estimates the probability density function (PDF) of a dataset with n data points. Each data point in the dataset is assigned a kernel $K(x)$ with the bandwidth h. Here, the same bandwidth is chosen for all points which depends on the density of the dataset. The kernel itself has to be a valid PDF. By taking the sum of all kernels at the point x, the PDF is estimated resulting in

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h^k} K(\frac{x - x_i}{h}).$$

The Gaussian kernel

$$K(x) = \frac{1}{(2\pi)^{-\frac{k}{2}} |\Sigma|^{-\frac{1}{2}}} \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right)$$

with the covariance matrix set to the identity matrix $\Sigma = I_k$ is used as kernel function. The value of the bandwidth has a huge impact on the resulting PDF. If it is chosen too small, the resulting estimation will overfit the problem; if the chosen bandwidth is too large, underfitting will occur. Further information on the KDE is available, e.g., by Murphy [13].

B. Anomaly Detection

For both algorithms, the detection of abnormal behaviour is defined in the following way. First, the normal model is estimated using only data with normal behaviour. For each model, the minimum log-likelihood for the training data is calculated. The log-likelihood is the natural logarithm of the likelihood function which is defined as the conditional probability that an outcome is generated by a specific parameter set. The minimum value will be the boundary between normality and anomaly. By Using only normal data for training, which is diverse enough to resemble at least most of the possible normal behaviour, it can be expected that abnormal data will generate a lower log-likelihood.

Each new data point will now be decided on by evaluating the normal model with the new data. The result will be the log-likelihood of the new data point. If it is lower than the boundary, the new point is considered as an anomaly.

C. Metrics

As previously mentioned, precision, recall and f1-score are used as metrics to evaluate the performance of the different algorithms. The precision is defined as

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}},$$

whereas the recall is defined as

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}.$$

The f1-score is the harmonic mean of the precision and recall. It is defined as

$$\text{f1-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

The recall describes the fraction of the positives which are actually classified as positive (true positives). Thus, a small recall means, that there are lots of false negative classifications. The precision describes the fraction of all positively classified results which are actually positives. Hence, a small precision equals a great number of false positive classifications. These metrics are commonly used in classification tasks. For an optimal classifier, the value of each metric should be "1". The used metrics are further explained, e.g., by Manning et al. [14].

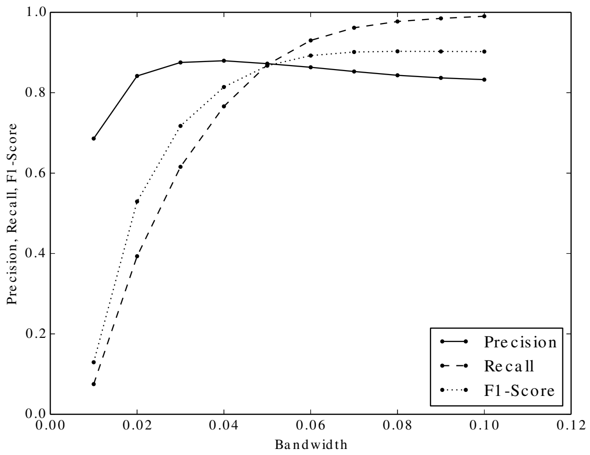


Fig. 5. Precision, recall, and f1-score for different bandwidths in the Fehmarn area.

TABLE I. OPTIMAL PARAMETER

	KDE	GMM	
Area	Bandwidth	# Components	Grid Size
Fehmarn	0.06	75	5x5
Kattegat	0.09	50	3x3

V. EMPIRICAL EVALUATION

For the empirical evaluation, the optimal parameters for both algorithms have to be determined. Therefore, a k-fold cross-validation as, e.g., described by Witten and Frank [15] is conducted for each parameter combination. The parameters to estimate are the bandwidth for the KDE and the number of components as well as the optimal grid-size for the GMM.

For the cross-validation, the dataset is first divided into normal and abnormal data. These two sets are then further divided into k parts, called folds, of equal size. In each step of the cross-validation, the model is trained with $k - 1$ folds of the normal data. The validation is performed by using the remaining fold together with one fold of the abnormal data. Therefore, the normal model is learned by using only normal data, and the validation can take place with the annotated data consisting of normal and abnormal data points. The results for the optimal parameters using a 3-fold cross-validation are shown in Table I. For each step in the cross-validation, the precision, recall, and f1-score are calculated. Finally, the cumulated means of these scores are determined.

Exemplarily, Fig. 5 shows precision, recall, and f1-score depending on the chosen bandwidth for the vessels in the Fehmarn area. For the KDE, the grid-size is not an important parameter to optimize. Due to the main principle behind the

TABLE II. SIMULATION RESULTS

	Fehmarn				Kattegat			
	KDE		GMM		KDE		GMM	
	Normal	Anomaly	Normal	Anomaly	Normal	Anomaly	Normal	Anomaly
Precision	0.9388	0.5620	0.9538	0.5414	0.9602	0.3844	0.9710	0.4744
Recall	0.9312	0.5926	0.9513	0.5549	0.9569	0.4040	0.9634	0.5342
F1-Score	0.9350	0.5769	0.9526	0.5481	0.9585	0.3940	0.9672	0.5025

KDE, only data points from the training set which are close to the evaluated data point will have an influence on the resulting probability density function. If the distance between data points from the training set and the evaluated data point is large, the resulting value of the kernel function will tend to zero. Thus, omitting some points will only decrease the calculation time, which is not important for this work.

To assess the performance of the algorithms, two different aspects are mainly considered. First, the performance to resemble the true normal model is estimated. This will give an idea about how well the true PDF is replicated by the used algorithms. The second is to identify abnormal behaviour. This will offer valuable insight about the performance of modelling abnormal behaviour by using training data consisting solely of normal data. Therefore, in Table II, precision, recall and f1-score for both are given.

The performance to resemble the normal model is quite high. The f1-score for the KDE is at 0.9350 for the Fehmarn area respectively 0.9585 for the Kattegat area. For the GMM, these values are at 0.9526 and 0.9672 respectively. Thus, both algorithms estimate the PDF of the normal behaviour well.

The identification of anomalies performed poorly compared to the normal model. The f1-score equals 0.5769 and 0.3940 for the KDE in the Fehmarn and the Kattegat area respectively. The GMM scored 0.5481 and 0.5025 for the f1-score, and thus 5% lower in the Fehmarn area and 27.5% higher in the Kattegat area. The precision score for the normal model is always higher than the recall, whereas this relation is reverted for the anomaly detection capability.

The difference between precision and recall for the anomaly detection is higher than the one for the normality detection. For the normality detection, precision and recall are more or less the same. The difference is less than 1%, and the precision is always higher than the recall. For the anomaly detection, the differences are greater (up to 12% higher), and the recall is always higher than the precision.

In Fig. 6 results regarding the Fehmarn area, and in Fig. 7 and Fig. 8 results for the Kattegat area are shown. For the Fehmarn area, plots of all folds are shown. These plots consist of a map of the area, grey lines representing the relevant traffic in the area and several dots. These dots resemble some of the tested points in each fold. The green dots represent the data points correctly found as anomalies, the red ones represent data points which are marked as normal despite being abnormal, and blue dots represent data points which are falsely marked as anomaly. Normal data points classified as normal are not depicted in this figure. The same colour-encoding applies for Fig. 7 and Fig. 8. In these figures, only the first fold, which contains for each algorithm identical data points, is shown.

The result figures for both areas show some general problems with the detection of anomalies. First of all, if a trajectory is quite near to the normal behaviour, and if the speed and heading of the vessel is similar to the normal model, the trajectory will not be identified as anomaly. This can be seen particularly in Fig 6(a). Another problem can be seen, when using a grid to divide the area as depicted in Fig. 6(b), Fig. 6(d), Fig. 6(f) and Fig. 8. If there is not enough data in one of the grid-cells to build a normal model, there will be no valid model for the decision about abnormal behaviour. Therefore, a different strategy has to be chosen; e.g., all data points in the cell are anomalies, or there are no anomalies in the cell. Here, all points in those cells are marked as anomaly.

Another problem will occur, if the position is normal, but either the heading, or the speed, or both are abnormal. In these cases, both algorithms perform poorly as, e.g., shown in Fig. 6(d). Here, in the cell 3x3, a vessel is driving in a different direction than the other vessels in this region (depicted by red dots), but is not detected as anomaly.

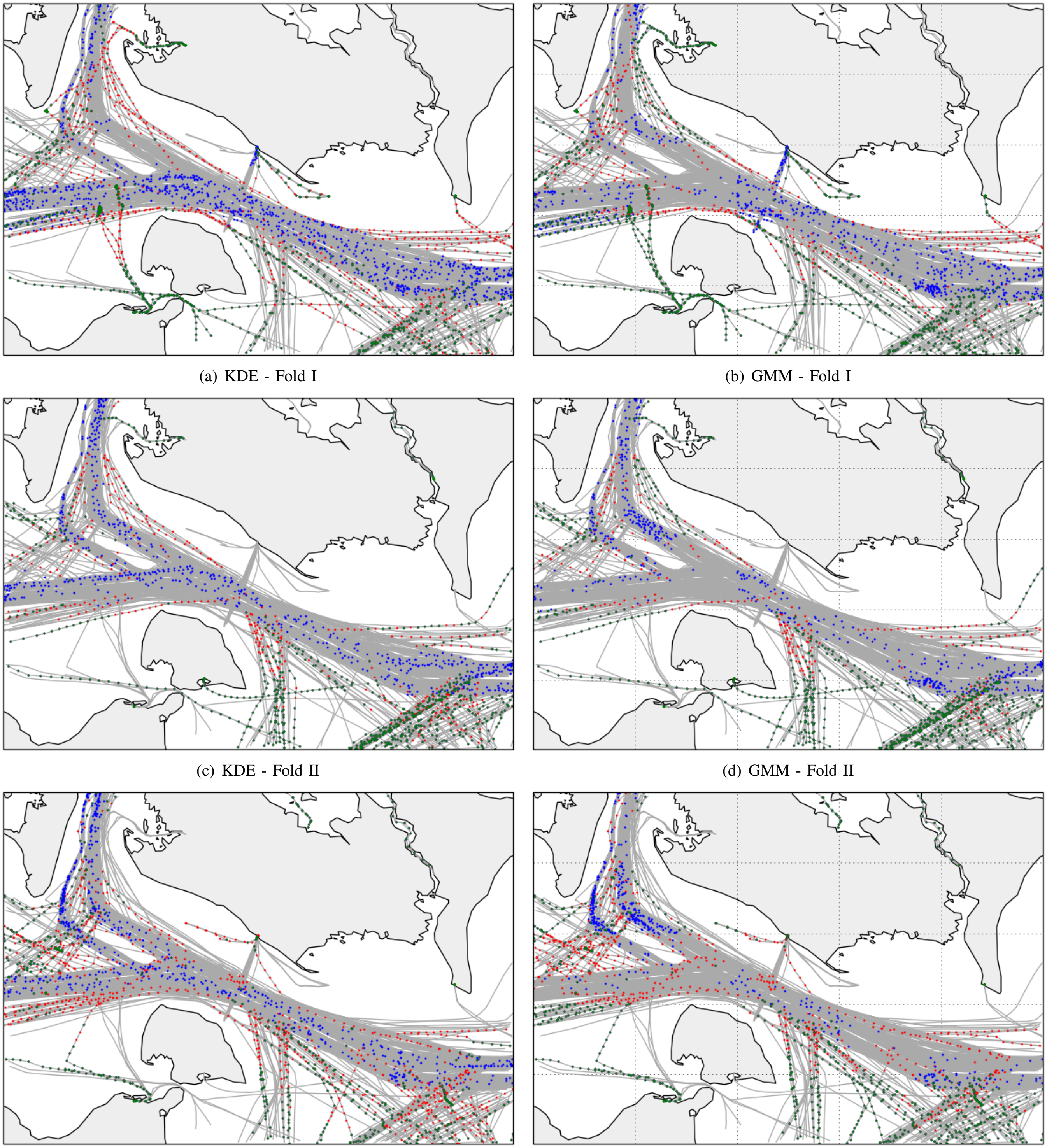
Between Fehmarn and Lolland, there is a ferry lane. Therefore, there is traffic between the two islands. While this is a normal behaviour, which repeats itself more or less often, the algorithms have problems to identify this as normal. This can be seen in Fig. 6(a) and Fig. 6(b). In these figures, only the part of the trajectory which crosses the main sea lane is not detected as anomaly. Particularly, the data around the harbour of Lolland is falsely detected as anomaly.

Furthermore, there are many falsely classified anomalies in both areas with both algorithms (blue dots). For the KDE, these occur evenly on all normal trajectories. For the GMM, the false classification depends mainly on the observed cell: In some cells, the amount of false anomaly classification is higher than in others; e.g., in Fig. 6(b), the cell 3x3 has more false anomalies than the cell 3x2.

VI. CONCLUSION

The amount of false detections as well as the amount of anomalies which are not found is high for both algorithms. Therefore, the question arises, whether these methods are actually able to reconstruct the real underlying PDF. In this regard, the GMM has a disadvantage for relying on multivariate normal distributions for each component. The actual distribution is unknown, and can be of another kind. A KDE is able to estimate an arbitrary PDF, if the amount of training data is great enough.

For both algorithms only a single point at a time is considered for evaluation, and not the whole trajectory. Hence, an anomaly as depicted in Fig. 9 will probably not be found,



(a) KDE - Fold I

(b) GMM - Fold I

(c) KDE - Fold II

(d) GMM - Fold II

(e) KDE - Fold III

(f) GMM - Fold III

Fig. 6. KDE (a, c, e) and GMM (b, d, f) results for the Fehmarn area for the different folds. The light grey polygons represent landmasses, the grey lines correspond to all trajectories, the dotted lines represent the used grid, the dots represent some evaluated data points. Green dots represent correctly found anomalies, red dots missed anomalies and blue dots normal points which are falsely declared as anomaly. The figures next to each other are from the same fold, and thus the same data is used to estimate and validate the models.

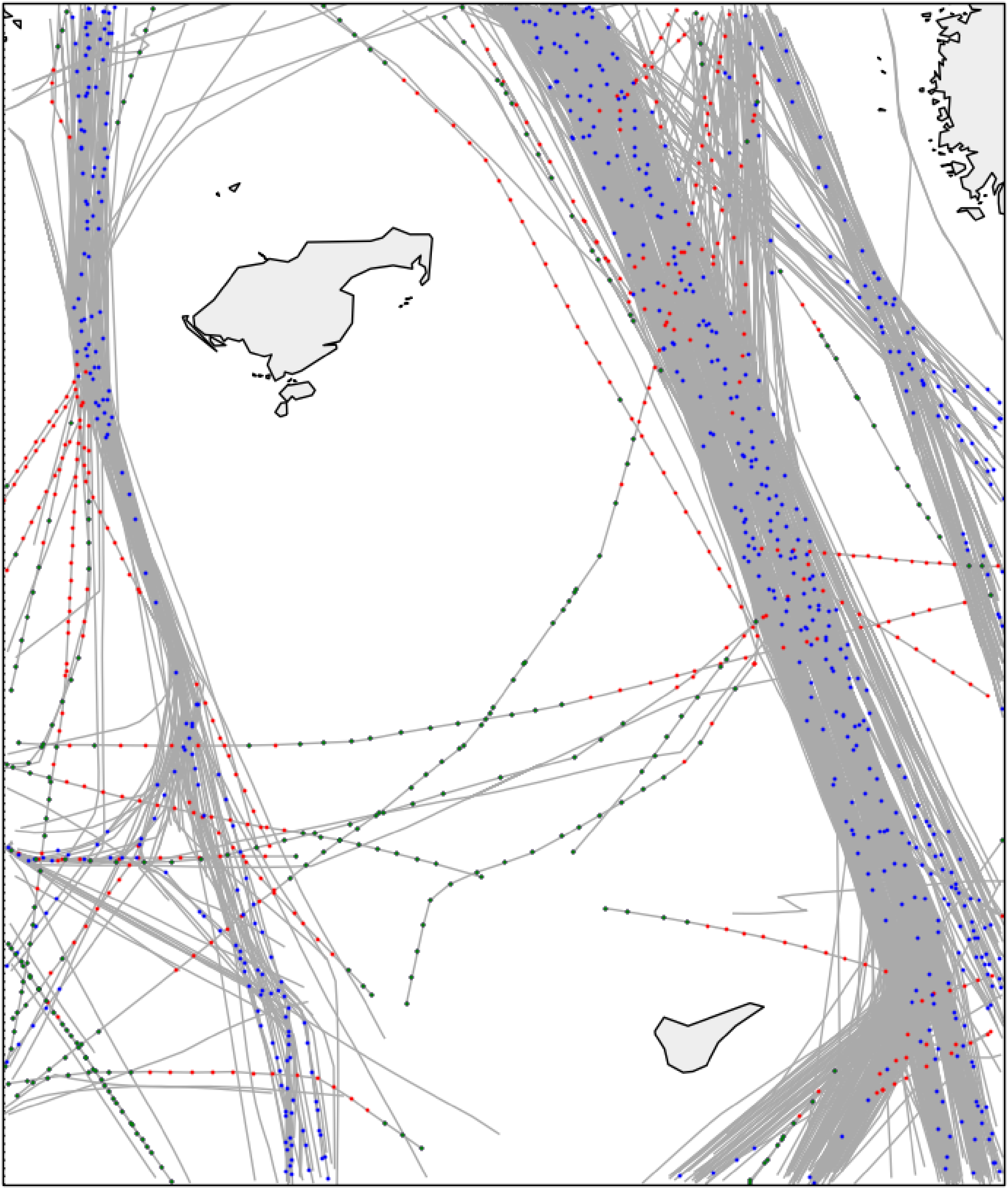


Fig. 7. KDE results for the Kattegat area for one fold. The light grey polygons represent landmasses, the grey lines correspond to all trajectories, the dots represent some evaluated data points. Green dots represent correctly found anomalies, red dots missed anomalies and blue dots normal points which are falsely declared as anomaly. The area is not further divided by using a grid.

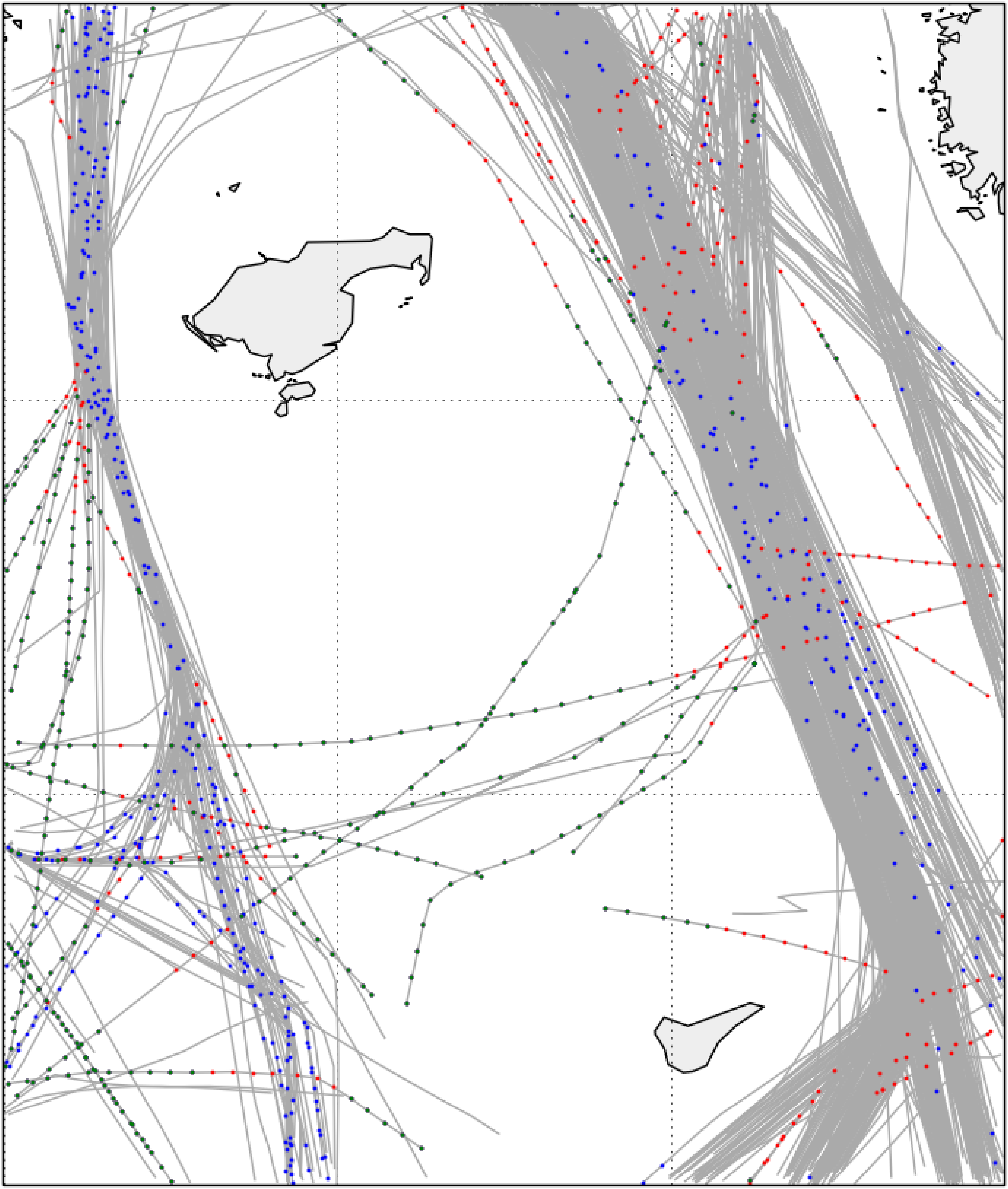


Fig. 8. GMM results for the Kattegat area for one fold. The light grey polygons are landmasses, the grey lines correspond to all trajectories, the dotted lines represent the used grid, the dots represent some evaluated data points. Green dots represent correctly found anomalies, red dots missed anomalies and blue dots normal points which are falsely declared as anomaly.

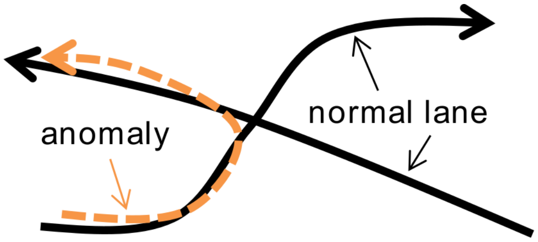


Fig. 9. Problem with point only evaluation. Two trajectories (each is only valid in the direction of the arrow) are crossing. An abnormal behaviour is depicted as an orange dashed line. It starts on one trajectory and changes to the other during the crossing. Depending on the context, this can be considered as anomaly.

because each point for itself is of normal behaviour. Only the combination of all points in the trajectory can be considered an anomaly.

The usage of an arbitrary grid to divide the area can yield problems. Particularly, in the border region between two cells, the grid can perform worse than using no grid at all. The aim of the EM algorithm is to fit the components of the GMM to the underlying data. Therefore, for each cell, the enclosed data is analysed separately. For each cell, the data abruptly ends at the border. Hence, it is likely to be less dense at the border compared to the center of a cell. Thus, the components of a trained GMM will more likely be placed in the center of the cell instead of the border, even though globally observed the data might have the same density at the center and at the border. Furthermore, if the grid was chosen unfavourably, the resulting model will not be able to learn a sea lane properly, because the grid might divide a sea lane or cut out parts of a sea lane.

Depending on the procedure to handle cells with not enough data to build a normal model, this can also result in worse performance. E.g., every point in these areas is marked as anomaly, then the grid can cut out a part of a lane with normal behaviour. If this area has not enough points, it will be detected as abnormal behaviour, even though it is not.

VII. Future Work

Even though, the optimal parameters are estimated, the same parameters are used for all grid-cells. Therefore, an improvement could be achieved by estimating the model parameters for each cell separately, respectively to use an adaptive approach for the bandwidth estimation for the KDE. Thus, the difference in density and complexity of each local area would be taken into account.

Currently, only quite simple algorithms for the anomaly detection are examined. The performance of these algorithms was suboptimal. Therefore, the next step is to compare more sophisticated algorithms. These algorithms should consider past points of a track while evaluating a new point. By incorporating the additional information provided using whole or partial trajectories, the results should improve compared to the density estimation using a GMM or a KDE.

Another open point is the annotation of the whole dataset and not only some artificial subsets. Currently, only a small subset of the whole dataset is annotated and inspected during the evaluation. By using the whole dataset, a better overview of the observed area can be used to get a better understanding of the normal behaviour, and thus to improve the models. Due to the greater amount of data, the methods of annotating the data must be reconsidered and improved. E.g., to achieve better model results the tracks could be annotated by domain experts or by using another strategy to ensure a consistent and reliable annotation. Also, using sea-maps to gain a better understanding of the sea lanes, shoals etc. will help to improve the annotated ground truth.

Acknowledgment

The underlying projects to this article are funded by the WTD 81 of the German Federal Ministry of Defense. The authors are responsible for the content of this article.

References

[1] Y. Fischer and J. Beyerer, “Ontologies for probabilistic situation assessment in the maritime domain,” in *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2013 IEEE International Multi-Disciplinary Conference on*, Feb 2013, pp. 102–105.

[2] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[3] B. Morris and M. Trivedi, “A survey of vision-based trajectory learning and analysis for surveillance,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 8, pp. 1114–1127, Aug 2008.

[4] R. Laxhammar, G. Falkman, and E. Sviestins, “Anomaly detection in sea traffic - a comparison of the gaussian mixture model and the kernel density,” in *Information Fusion (FUSION), 2009 12th International Conference on*, 2009.

[5] C. Brax and L. Niklasson, “Enhanced situational awareness in the maritime domain: an agent-based approach for situation management,” in *Proc. SPIE*, vol. 7352, 2009, pp. 735 203–735 203–10.

[6] M. Andersson and R. I. Johansson, “Multiple sensor fusion for effective abnormal behaviour detection in counter-piracy operations,” in *Waterside Security Conference (WSS), 2010 International*, Nov 2010, pp. 1–7.

[7] R. Laxhammar and G. Falkman, “Sequential conformal anomaly detection in trajectories based on hausdorff distance,” in *Information Fusion (FUSION), 2011 14th International Conference on*, July 2011, pp. 1–8.

[8] N. L. Guillarme and X. Lerouvreur, “Unsupervised extraction of knowledge from s-ais data for maritime situational awareness,” in *Information Fusion (FUSION), 2013 16th International Conference on*, July 2013, pp. 2025–2032.

[9] Y. Fischer, A. Reiswich, and J. Beyerer, “Modeling and recognizing situations of interest in surveillance applications,” in *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2014 IEEE International Inter-Disciplinary Conference on*, March 2014, pp. 209–215.

[10] G. Shafer and V. Vovk, “A tutorial on conformal prediction,” *Journal of Machine Learning Research*, vol. 9, pp. 371–421, March 2008.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[12] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2014.

[13] K. P. Murphy, *Machine learning: a probabilistic perspective*. Cambridge, MA: MIT Press, Cambridge, MA, 2012.

[14] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[15] I. H. Witten and E. Frank, *Data Minig: Practical Machine Learning Tools and Techniques with Java Implementations*, 2nd ed. Morgan Kaufmann Publishers, 2005.